A Survey in Applications of Deep Learning in GIS – Spatiotemporal data mining and forecasting

Prepared by

Supasin Wuthikulphakdi

ID# 608785068

Tamkang University

October 8, 2021

A proposal submitted in partial fulfillment of the requirements for the Master Degree of Science in Computer Science

Proposed Supervisor:

……………………………………………………………………………

Isaac TSAI Yihjia Ph.D

Proposed Reader(s):

……………………………………………………………………………

LEE, Jian-Zong

Faculty of Engineering

Faculty of Engineering – Department of Computer Science

Date of Submission: October 8, 2021

A Survey in Applications of Deep Learning in GIS – Spatiotemporal data mining and forecasting

Prepared by

Supasin Wuthikulphakdi

ID# 608785068

Tamkang University

October 8, 2021

A proposal submitted in partial fulfillment of the requirements for the Master Degree of Science in Computer Science

Proposed Supervisor:

………………………………………………………………………………

Isaac TSAI Yihjia Ph.D

Proposed Reader(s):

………………………………………………………………………………

LEE, Jian-Zong

Faculty of Engineering

Faculty of Engineering – Department of Computer Science

Date of Submission: October 8, 2021

CHAPTER1: INTRODUCTION & BACKGROUND

1.1 Overview, Background & Purpose

This thesis is to explore how deep learning architectures and models can be used in ST-data mining in traffic flow prediction, intelligent transport system (ITS) and weather/disaster forecast. And to discover what group of models, to date, is the best for STDM, especially in traffic flow prediction – such as ST-LSTM, and LT-LSTM.

Nowadays, data mining is a powerful analytic tool. As some forms of static datasets (i.e., images in the form of matrices) and sequential ones (i.e., time series, Natural Language Processing (NLP), etc.) are being mined to train Machine Learning (ML) analytic and predictive models.

As defined in [17], spatiotemporal data is a dataset relate to both space and time. It is collected in the space domain (a.k.a. a (heat) map over a location in a form of raster and vector data matrices), and the time domain (a.k.a. the time series - a heatmap average for each timestep in the form of timestep-D vectors, also includes NLP datasets). Overall, when concatenated, a spatiotemporal dataset is usually a 3D+ tensor.

Given the popularity of GPS-supporting devices, including smartphones, the need of S.T.-data mining rises with the emerging use of intelligent transport system (ITS, i.e., automatic guidance, self-driving cars, traffic prediction, etc.), and disaster prediction (i.e. weather, flood, earthquake, storms, clouds, smog, global warming etc.).

To manually collect, process, and forecast the ST data is a labourious task. Several models based on both machine learning have been deployed, but ML models need a human to extract the feature representations. Deep learning models, capable of self-feature extraction, usually outperform regular ML ones. Fundamentally, a ST-Deep Learning model works as follows:

- A CNN learns the spatial input (i.e., heatmap, graph data (GCN), & data coordinates)
- While and LSTM/RNN learns the temporal domain of the input data.
- With both DNN working together, sometimes with an intermediate submodel (i.e., a dense submodel in [14]), we can correlate and forecast a spatiotemporal data (such as traffic, train ridership, etc.) not only the time series (time domain), but also the heatmap (space domain).

However, DL models have drawbacks, a CNN/GCN/FCN can only extract and learn only the space domain, while an RNN/LSTM/seq2seq can only do these on the temporal domain of the S.T. data. Proven by several papers, RNN, when processing a long sequence, has a problem of vanishing/exploding gradient. Thus, LSTM is preferred in later works

As the result, some work referred in the LR section overcame the problems by adding such two neural networks into the same framework, allowing them to extract and learn the ST-data of both domains simultaneously. More technical details can be read in [8] and [19].

My thesis is focused on paper survey of deep learning algorithms and models in ST-data mining & forecasting. With the AI, there are some uses in spatiotemporal data mining:

- CNN-GCN-FCNs are used to extract features in the spatial domain (i.e. heatmap).
- LSTM-RNN-seq2seq are used to extract features in the temporal domain (i.e., sequence type data).
- In survey engineering, some researchers use CNN observe changes in land use, crop growth, and construction progresses.
- In ITS, ST-Data mining is extremely useful in traffic prediction, guidance systems, route planning, and self-driving cars.
- In Meteorology, the STDM is used for weather and disaster forecast.
- To make GPS based pathfinding more accurate, Spatiotemporal Data Mining is often utilized in several papers to train the pathfinding ML/DL-architectures.
- In transportation, deep learning methods learn highly intricate ST-correlations among the traffic data – useful in some tasks such as traffic flow prediction, traffic incident detection, and traffic congestion prediction – such as in [13] and [14].
- In On-demand service, such as Uber [10], to perform pathfinding
- All the papers to be referred below, may be added & updated in the future, involve GIS, Computer Vision (CV), time series prediction, and spatiotemporal data mining.

1.2 Outline & Technical Concepts

According to [8], there are some fundamentals of STDM we need to know before proceeding any further, for example:

- Spatiotemporal Data Structures
- Data Instances
- Data Respresentations
- Deep Learning Models in STDM

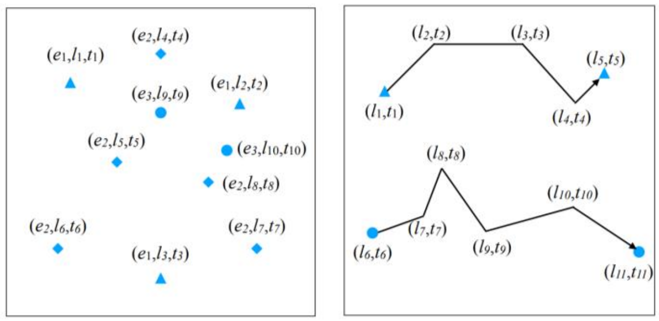
1.2.1 Data instances.

Usually, a spatiotemporal dataset is a 3D tensor indicating space (like a book page) and time (number of pages). In general, the ST data can be summarized into the following data instances: points, trajectories, time series, spatial maps and ST raster as shown in the left part of Fig. 4. A ST point can be represented as a tuple containing the spatial and temporal information as well as some additional features of an observation such as the types of crimes or traffic accidents. Besides ST events, trajectories and ST point reference can also be formed as points. For example, one can break a trajectory into several discrete points to count how many trajectories have passed a particular region in a particular time slot.

Different data instances can be extracted from ST raster as time series, spatial maps or ST raster itself, depending on different applications and analytic requirements. First, we can consider the measurements at a particular ST grid of the ST field as a time series for some time series mining tasks. Second, for each time stamp the measurements of an ST raster can be considered as a spatial map. Third, one can also consider all the measurements spanning all the locations and time stamps as a whole for analysis. In such a case, ST raster itself can be a data instance.

1.2.1.1 Event data.

Event data comprise of discrete events occurring at point locations and times (e.g., crime events in the city and traffic accident events in a transportation network). An event can generally be characterized by a point location and time, which denotes where and when the event occurred, respectively. For example, a crime event can be characterized as such a tuple (ei , li , ti), where ei is the crime type, li is the location where the crime occurs and ti is the time when it occurs.



(a) Three types of events

(b) The trajectories of two moving objects

Fig. 1. Illustration of event and trajectory data types

1.2.1.2 Trajectory data

Trajectories denote the paths traced by bodies moving in space over time. (e.g., the moving route of a bike trip or taxi trip). Trajectory data are usually collected by the sensors deployed on the moving objects that can periodically transmit the location of the object over time, such as GPS on a taxi. Fig. 1(b) shows an illustration of two trajectories. Each trajectory can be usually characterized as such a sequence {(l1, t1),(l2, t2)...(ln, tn)}, where li is the location (e.g. latitude and longitude) and ti is the time when the moving object passes this location.

1.2.1.3 Point reference data.

Point reference data consist of measurements of a continuous ST field such as temperature, vegetation, or population over a set of moving reference points in space and time. For example, meteorological data such as temperature and humidity are commonly measured using weather balloons floating in space, which continuously record weather observations. Point reference data can be usually represented as a set of tuples as follows {(r1, l1, t1),(r2, l2, t2)...(rn, ln, tn)}. Each tuple (ri , li , ti) denotes the measurement of a sensor ri at the location li of the ST filed at time ti .

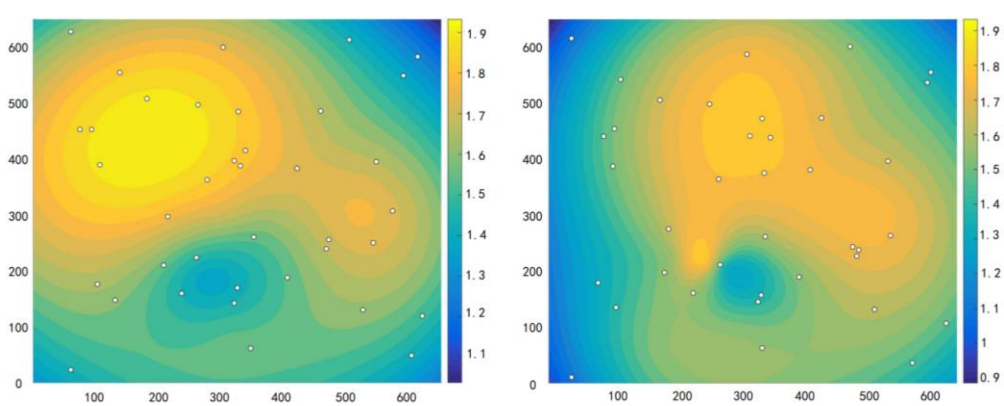


Fig. 2. Illustration of ST reference point data in two timestamps. The while circles are the locations of the sensors that record the readings of the ST field. The color bars show the distribution of the ST field.

1.2.1.4 Raster data.

Raster data are the measurements of a continuous or discrete ST field that are recorded at fixed locations in space and at fixed time points. The major difference between point reference data and raster data is that the locations of the point reference data keep changing while the locations of the raster data are fixed. The locations and times for measuring the ST field can be regularly or irregularly distributed. Given m fixed locations S = {s1, s2, ...sm} and n time stamps T = {t1, t2, ...tn}, the raster data can be represented as a matrix Rm×n, where each entry rij is the measurement at location si at time stamp tj . Raster data are also quite common in real world applications such as transportation, climate science, and neuroscience. For example, Fig. 3 shows an example of the traffic flow raster data of a transportation network. Each road is deployed a traffic sensor to collect real time traffic flow data. The traffic flow data of all the road sensors in a whole day (24 hours) form a raster data.

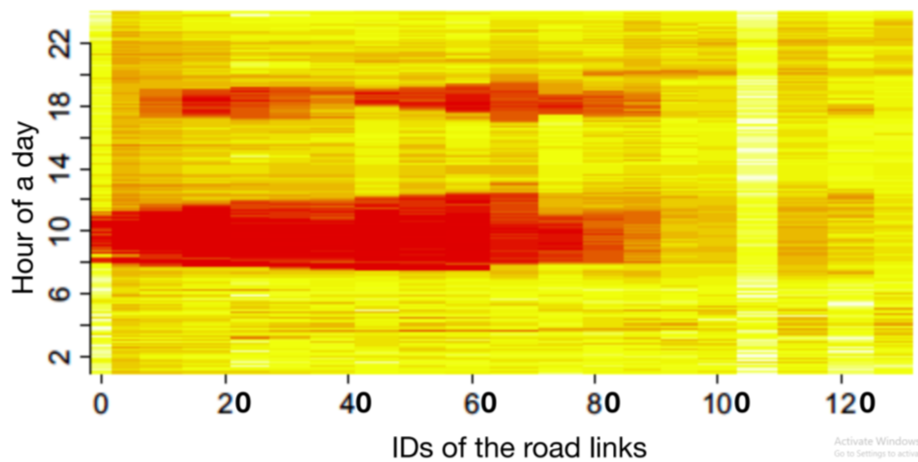


Fig. 3. Illustration of raster data collected from traffic flow sensors. The x-axis is the ID of the road links in a transportation network, and the y-axis is the hour of a day. Different colors denote different traffic flows on the road links captured by the road sensors deployed at fixed locations.

1.2.1.5 Video.

A video that consists of a sequence of images can be also considered as a type of ST data. In the spatial domain, the neighbor pixels usually have similar RGB values and thus present high spatial correlations. In the temporal domain, the images of consecutive frames usually change smoothly and present high temporal dependency. A video can be generally represented as a three-dimensional tensor with one dimension representing time t and the other two representing an image. Actually, video data can be also considered as a special raster data if we assume that there is a “sensor” deployed at each pixel and at each frame the “sensors” will collect the RGB values.

1.2.2. Data Representations.

For the abovementioned five types of ST data instances, four types of data representations are generally utilized to represent them as the input of various deep learning models, sequence, graph, 2-dimensional matrix, and 3-dimensional tensor as shown in the right part of Fig. 4. Different deep learning models require different types of data representations as input. Thus, how to represent the ST data instances relies on the data mining task under study and the selected deep learning model.

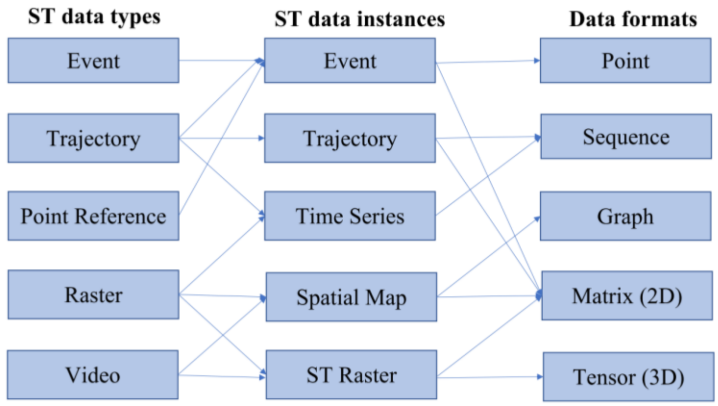


Fig. 4. Data instances and representations of different ST data types

1.2.3. Deep learning models.

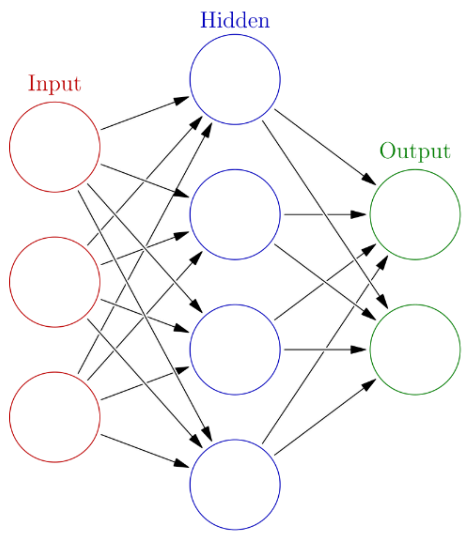


Fig.5 a diagram of an Artificial Neural networks

Deep learning models, praised for their outstanding performance and accuracy, are often utilized in several STDM papers. There are some notable examples below.

1.2.4.1 Restricted Boltzmann Machines (RBM).

A Restricted Boltzmann Machine is a two-layer stochastic neural network [53] which can be used for dimensionality reduction, classification, feature learning and collaborative filtering. As shown in Fig. 6, the first layer of the RBM is called the visible, or input layer with the neuron nodes {v1, v2, ...vn}, and the second is the hidden layer with the neuron nodes {h1, h2, ...hm}. As a fully connected bipartite undirected graph, all nodes in RBM are connected to each other across layers by undirected weight edges {w11, ...wnm}, but no two nodes of the same layer are linked. The standard type of RBM has a binary-valued nodes and bias weights. RBM tries to learn a binary code or representation of the input, and depending on the task, RBM can be trained in either supervised or unsupervised ways. RBM is usually used for learning features.

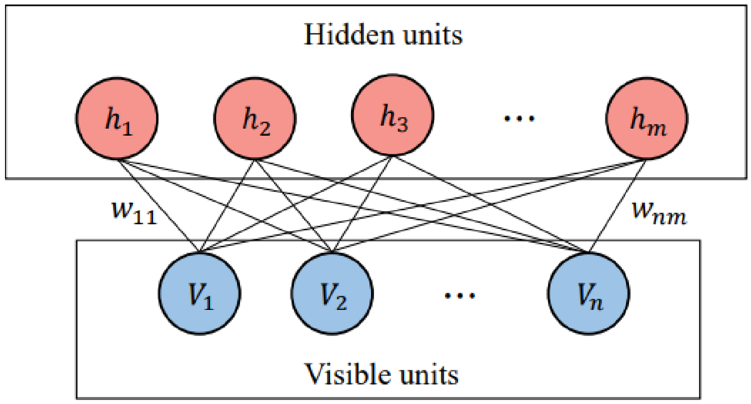


Fig. 6. Structure of the RBM model.

1.2.4.2 Convolutional Neural Networks (CNN).

Convolutional neural networks (CNN) is a class of deep, feed-forward artificial neural networks that are applied to analyze visual imagery. A typical CNN model usually contains the following layers as shown in Fig. 7: the input layer, the convolutional layer, the pooling layer, the fully connected layer and the output layer.

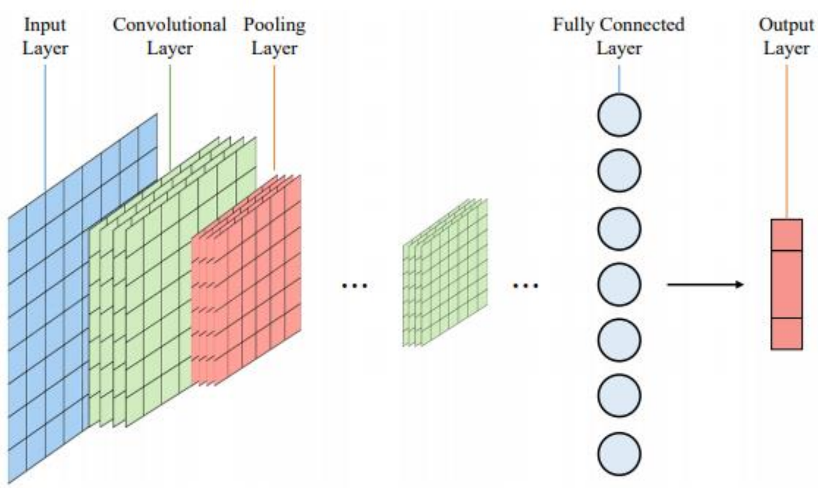


Fig. 7. Structure of the CNN model.

First, the convolutional layer will determine the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume. Second, the pooling layer will then downsample the spatial dimensionality of the given input to reduce the number of parameters. The fully connected layers will connect every neuron in one layer to every neuron in the next layer to learn the final feature vectors for classification.

CNN is designed to process image data. Due to its powerful ability in capturing the correlations in the spatial domain, it is now widely used in mining ST data, especially the spatial maps and ST rasters.

1.2.4.3. Graph CNN (GCNN).

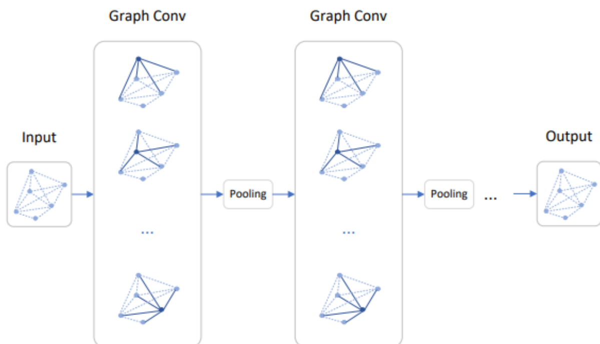


Fig. 8. Structure of GraphCNN model.

CNN is designed to process images which can be represented as a regular grid in the Euclidean space. Graph CNN is recently widely studied to generalize CNN to graph structured data [160]. Fig. 8 shows a structure illustration of a Graph CNN model. The graph convolution operation applies the convolutional transformation to the neighbors of each node, followed by pooling operation. By stacking multiple graph convolution layers, the latent embedding of each node can contain more information from neighbors which are multi hops away. After the generation of the latent embedding of the nodes in the graph, one can either easily feed the latent embeddings to feed-forward networks to achieve node classification of regression goals or aggregate all the node embeddings to represent the whole graph. And then perform graph classification and regression.

1.2.4.4 RNN and LSTM.

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. RNN is designed to recognize the sequential characteristics and use patterns to predict the next likely scenario. They are widely used in the applications of speech recognition and natural language processing. Fig. 9(a) shows the general structure of a RNN model, where Xt is the input data, A is the parameters of the network and ht is the learned hidden state. One can see the output (hidden state) of the previous time step t−1 is input into the neural of the next time step t. Thus, the historical information can be stored and passed to the future.

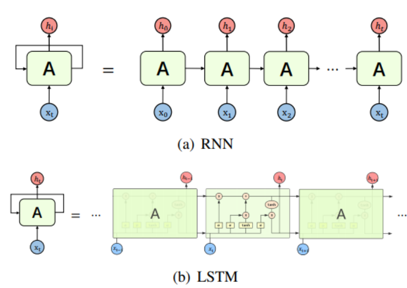


Fig. 9. Structure of the RNN and LSTM models

A major issue of standard RNN is that it only has short term memory due to the issue of vanishing gradients. Long Short-Term Memory (LSTM) network is an extension for recurrent neural networks, which is capable of learning long term dependencies of the input data. LSTM enables RNN to remember their inputs over a long period of time due to the special memory unit as shown in the middle part of Fig. 9(b). An LSTM unit is composed of three gates: input, forget and output gate. These gates determine whether to let new input in (input gate), delete the information because it is not important (forget gate) or to let it impact the output at the current time step (output gate). Both RNN and LSTM are widely used to deal with sequence and time serious data for learning the temporal dependency of the ST data.

1.2.4.5. Seq2Seq.

A sequence to sequence (Seq2Seq) model aims to map a fixed length input with a fixed length output where the length of the input and output may differ [138]. It is widely used to various NLP tasks such as machine translation, speech recognition and online chatbot. Although it is initially proposed to address NLP tasks, Seq2Seq is general framework and can be used to any sequence-based problem. As shown in Fig. 10, a Seq2Seq model generally consists of 3 parts: encoder, intermediate (encoder) vector and decoder. Due to the powerful ability in capturing the dependencies among the sequence data, Seq2Seq model is widely used in ST prediction tasks where the ST data present high temporal correlations such as urban crowd flow data and traffic data.

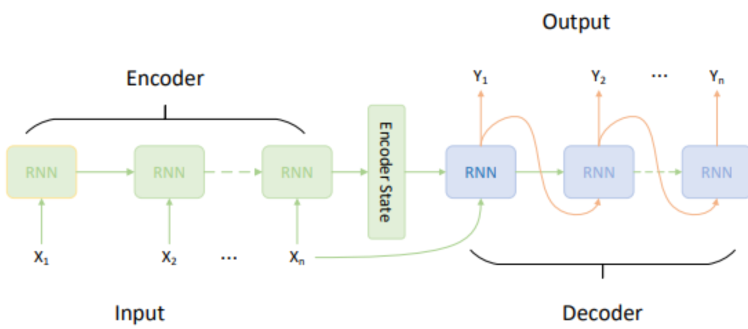


Fig. 10. Structure of Seq2Seq model.

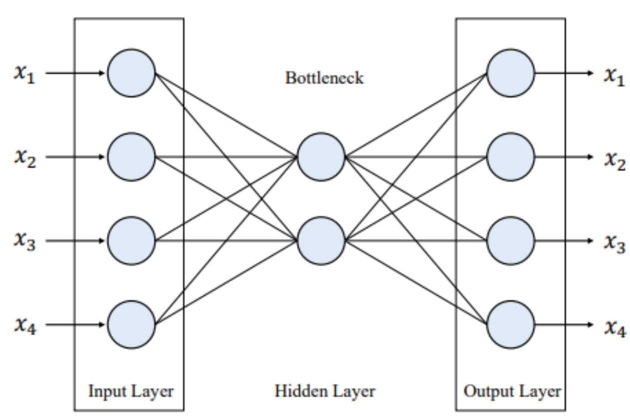


Fig. 11. Structure of the one-layer AE model.

1.2.4.6 Autoencoders (AE)

An autoencoder is a type of artificial neural network that aims to learn efficient data codings in an unsupervised manner [53]. As shown in Fig. 11, it features an encoder function to create a hidden layer (or multiple layers) which contains a code to describe the input. There is then a decoder which creates a reconstruction of the input from the hidden layer. An autoencoder creates a compressed representation of the data in the hidden layer or bottleneck layer by learning correlations in the data, which can be considered as a way for dimensionality reduction. As an effective unsupervised feature representation learning technique, AE facilitates various downstream data mining and machine learning tasks such as classification and clustering. A stacked autoencoder (SAE) is a neural network consisting of multiple layers of sparse autoencoders in which the output of each layer is wired to the inputs of the successive layer [7].

1.2.4.7 Attention Mechanisms (AM)

Attention is a mechanism that was developed to improve the performance of the Encoder-Decoder RNN on machine translation [5]. As an unattended Encoder-Decoder encodes only limited length sequence representation, it performs poorly with long input sequences. The solution of the AM allows the model to learn which encoded words in the source sequence to pay attention to and to what degree during the prediction of each word in the target sequence. It also works against ST-dataset too in the form of visual attention.

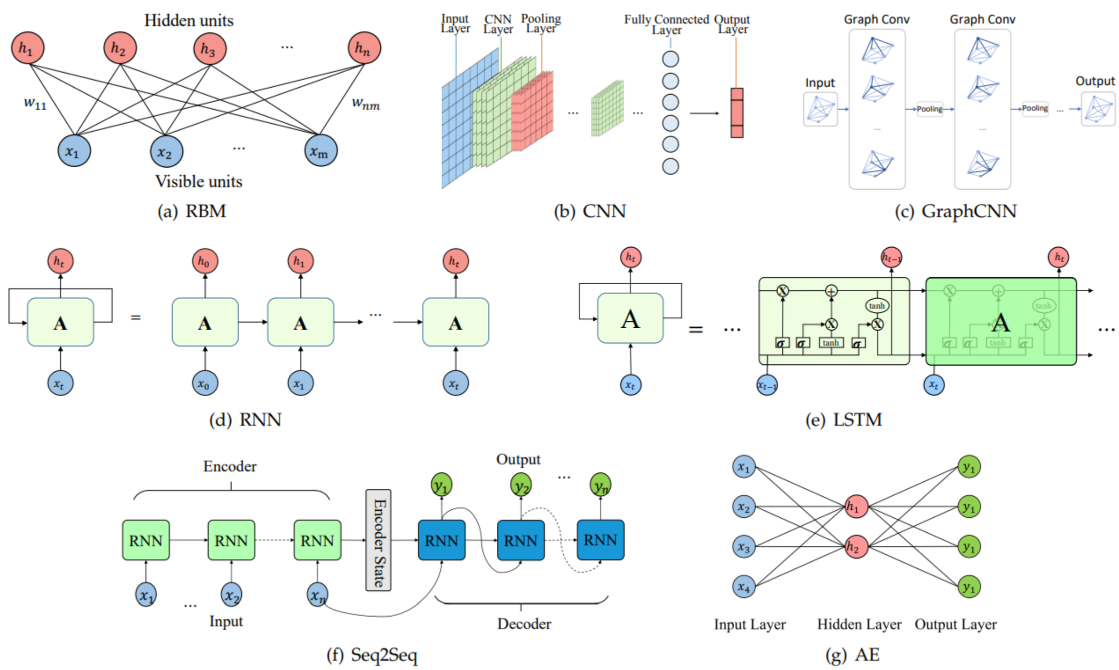


Fig.12 an overall diagram of an Artificial Neural Networks used in STDM

CHAPTER 2: LITERATURE REVIEW & RELATED WORKS

2.1 Notable referred works

Doshi, Basu and Pang [1] developed a CNN model identifying disaster-impacted areas by comparing the change in man-made features extracted from satellite imagery. Using a pre-trained semantic segmentation model from [2] they extracted man-made features, the pre- and post-event images on the "before, during and after" imagery of the event-affected area.

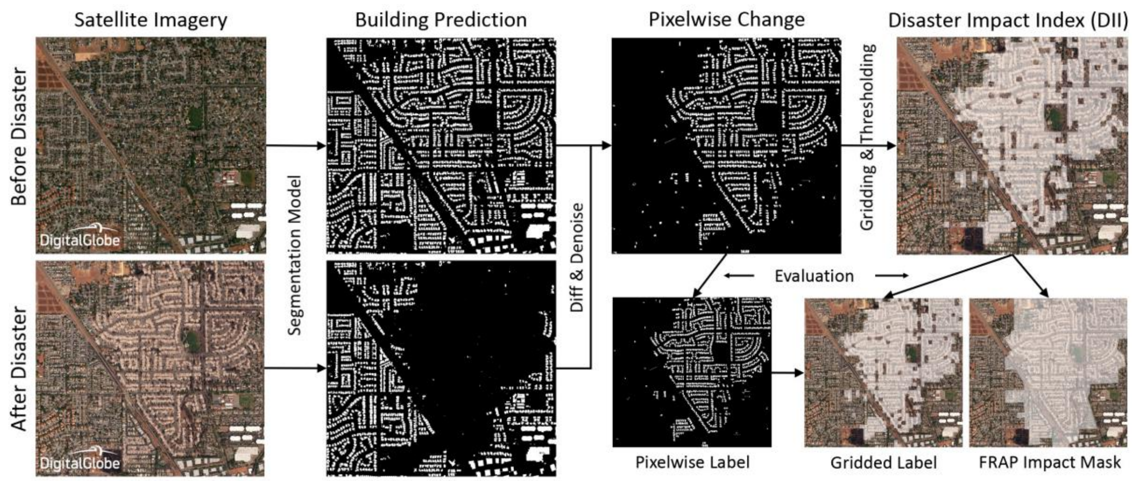


Fig 13. – Doshi's Residual Inception Skipnet model for disaster insight [1].

Amit and Aoki [3] proposed a CNN consists of sequence of layers, the convolution layer (who detects features from a data image), the pooling layer (downsamples the input), and the FC layer (who classifies the features detected earlier), with ReLU as the main activation function of the network. Further explained in the section 2 of [3].

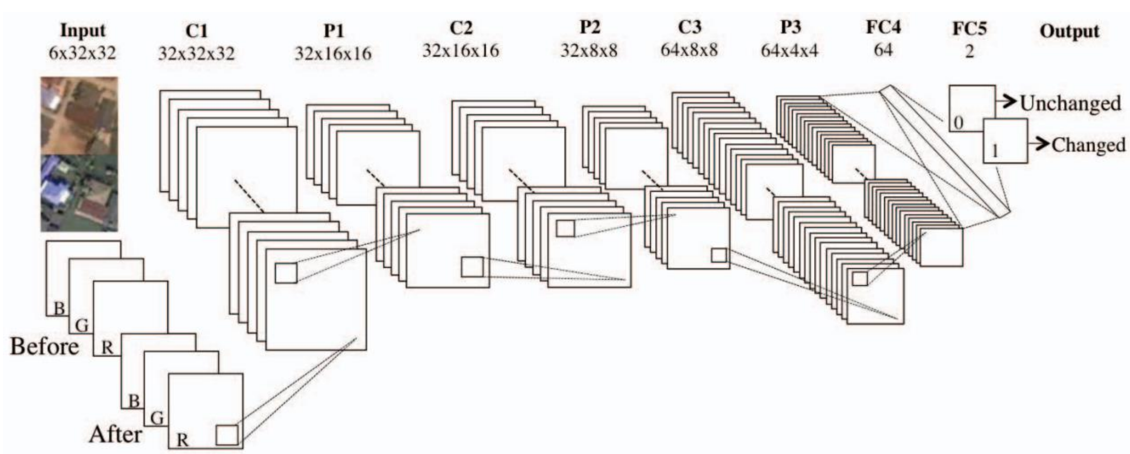


Fig 14. – Amit and Aoki's CNN based disaster detection model used in [3].

Iglovikov, Mushinskiy and Osin [4], used an FC-CNN named U-NET, along with an embedded multispectral sensor, which detects frequency reflection by the objects, to detect geo-features in satellite images and yielded satisfying results.

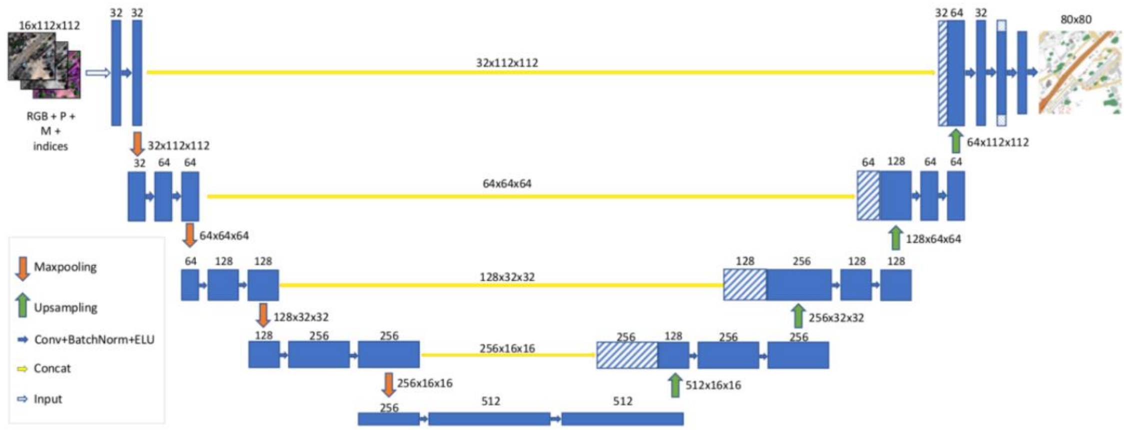


Fig 15. – Iglovikov's UNet architecture for geo-feature detection, featuring the downsampling and the upsampling sections [4].

Bochkovskiy, Wang and Liao [5] incorporated YOLO V.4, and TensorFlow Keras in CNN, to improve performance in image recognition. So, it is possible for us to deploy such a CNN model in this thesis. We hope that the deep learning models we can find, including our very own if we have time to design one, written in Python, will work as the goals above.

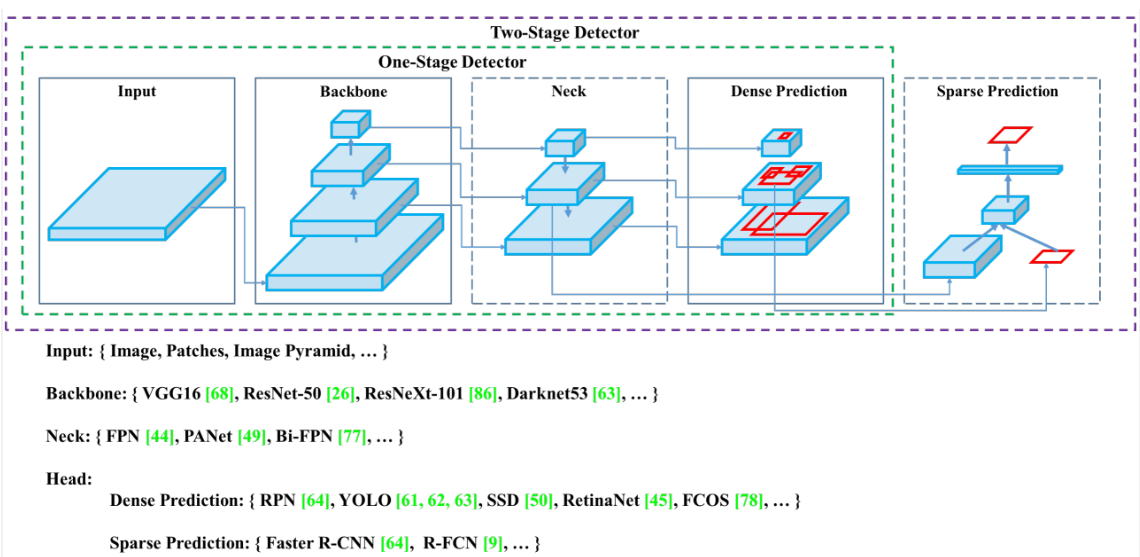


Fig 16. – Bochkovskiy's Yolov4 architectural diagram used in [5].

In Terms of video and sequence type photos (such as slideshow), however, the use of LSTM-RNN is needed. According to Fang et al. [6], LSTM is excellent at predicting flood because it could process time series data.

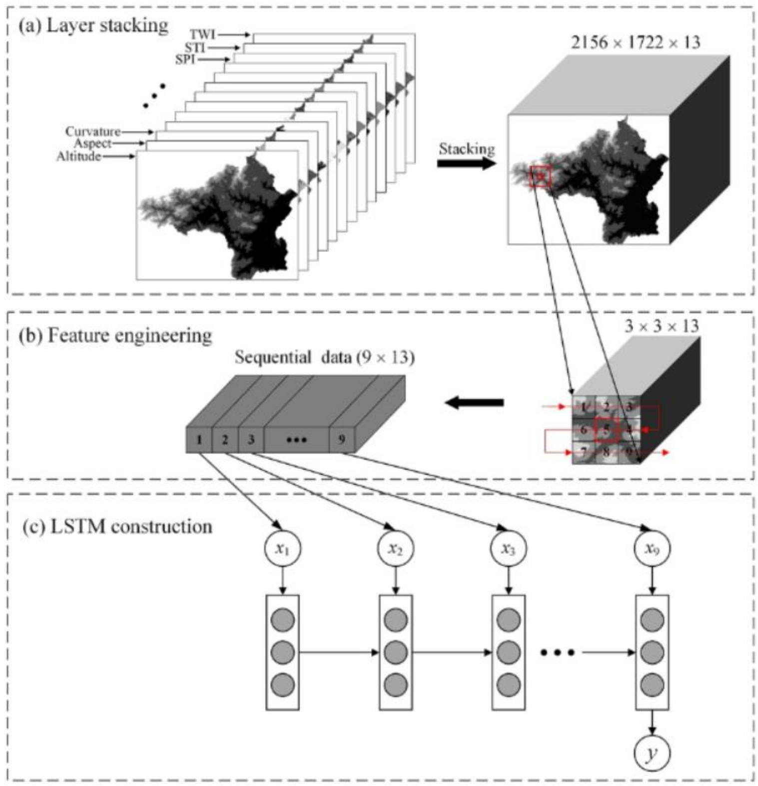


Fig 17. – Mechanics of Fang's LSTM model used in [6] to predict flood in Shangyou county, Jiangxi, China.

Li et al. [7], view spatiotemporal forecasting as a crucial task for a learning system that operates in a dynamic environment. It can be useful in pathfinding, autonomous vehicles, logistics, city planning etc. They used a Diffusion Convolutional Recurrent Neural Network (DCRNN) model to forecast the road traffic within a specific space and timeframe (The dataset was METR-LA, 2014). Diffusion convolution extracts the traffic features, and the RNN processes the traffic volumes in sequence.

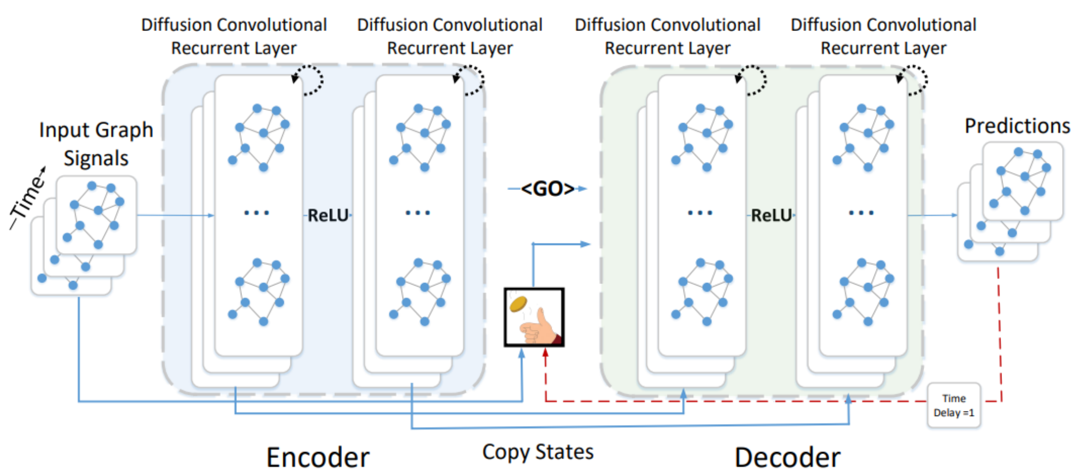


Fig 18. – Mechanics of Li's DRCNN model used in [7] to predict the traffic density of each timestamp and location.

Wang et al, 2019. [8], surveyed and collected several papers about spatiotemporal data mining. And explained the fundamentals and the concepts of STDM. According to the paper and fig 19, the majority of Deep learning STDM models are used for prediction, especially traffic prediction, which comprises the majority of the works referred by this paper.

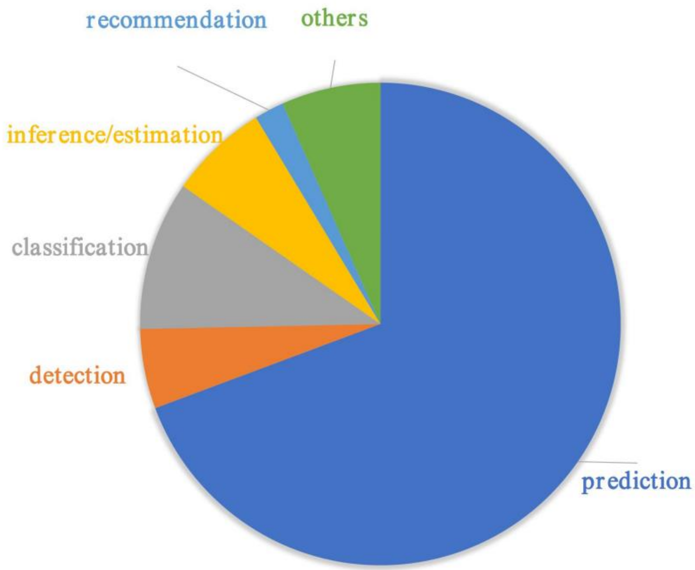


Figure 19: Distribution of the STDM problems addressed by deep learning

Yu et al, 2018. [9], proposed a Spatiotemporal Graph Convolutional Networks (STGCN), to tackle the time series prediction problem in traffic domain. They formulated the problem on graphs and build the model with complete convolutional structures, enabling much faster training speed with fewer parameters. Compared with existing models, STGCN more effectively captured comprehensive spatiotemporal correlations through modeling multi-scale traffic networks and consistently outperforms state-of-the-art baselines on various real-world traffic datasets.

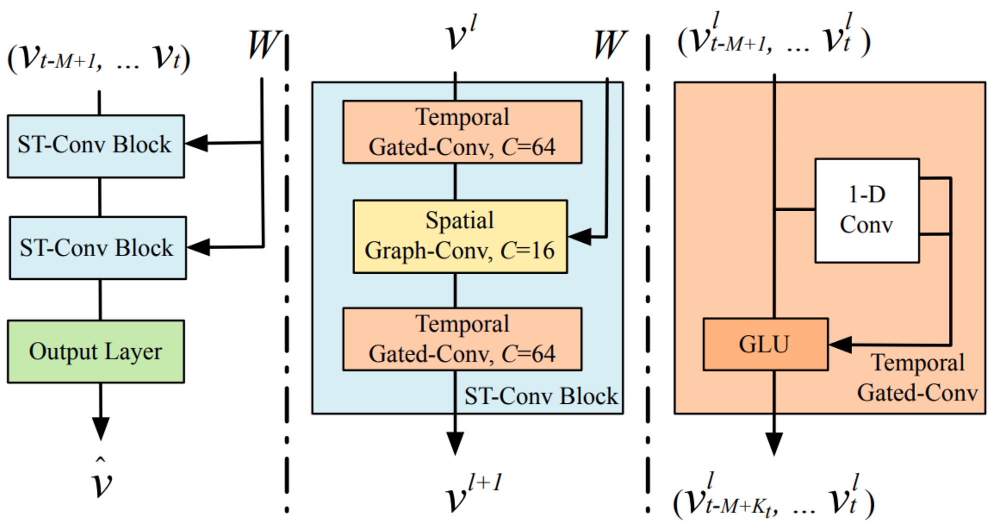
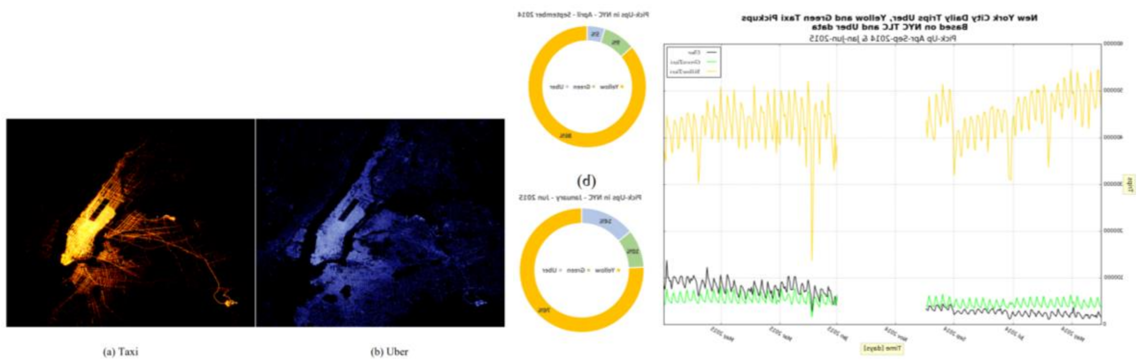


Figure 20: Yu's architecture of spatio-temporal graph convolutional networks – the full mechanism is described in the 3rd section of [9]

Correa et al, 2017 [10] performed a spatiotemporal data mining of Taxi vs Uber ridership in NYC, 2014+15. According to fig 21(a), the ridership for both taxi systems depended on several factors – such as personal income, education, jobs, car ownership etc. With 3 spatial models for ridership prediction – linear, spatial error and spatial lag models, the last one outperformed not only the first 2 algorithms, but also yielded a considerable accuracy and performance.



(a) (b)

Fig 21. – Correa's Taxi-Uber dataset visualization. The spatial module is displayed as heatmap (19a) which is used to predict hailing densities. Meanwhile, the temporal module is displayed as line graph (19b), which is used to predict future ridership volume.

Amato et al. [11] designed a deep learning-based architecture called "Empirical Orthogonal Functions principal component analysis" EOF-PCA in which the EOF framework decomposes the spatiotemporal input data, in terms of a sum of products of temporally referenced basis functions and of stochastic spatial coefficients which can be spatially modelled and mapped on a regular grid. Then, the input layer spatial covariates are processed by a "Fully Connected Neural Network" (FCNN) to obtain predictive coefficient to be recomposed altogether with the decomposed data stream, obtaining a spatiotemporal signal reconstruction.

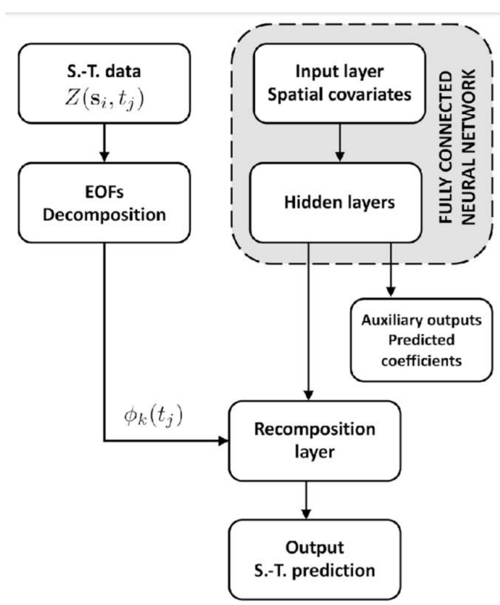


Fig 22. Amato's architecture. The temporal bases are extracted from a decomposition of the S.T. signal using EOFs. Then, an FCNN is used to learn the corresponding spatial coefficients [11].

Tang et al. [12] designed an LSTM based framework to learn and forecast the rail traffic. The short-term forecast of rail transit is an issue in intelligent transportation system (ITS). Accurate forecast can forewarn travel outburst, helping the passengers with their travel plans. Even though the LSTM is notably effective in temporal data, it cannot correlate the time domain with the space domain. That is why we propose ST-LSTM. Compared with other conventional models, ST-LSTM network can achieve a better performance in experiments.

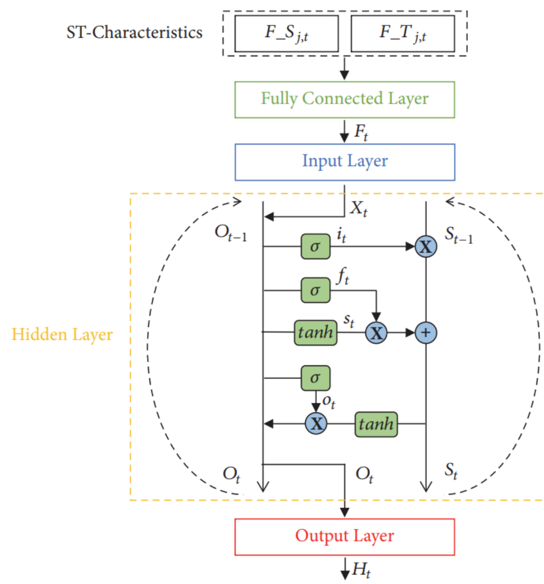


Fig 23. Tang's ST-LSTM architecture [12]

Lu et al. [13], designed a spatial-temporal deep learning network, termed ST-TrafficNet, for traffic flow forecasting, and whose architecture works as follows. 1. The Spatial Aware Multi-Diffusion Convolution Bloc (ADC-Block – who introduces Graph Attention Mechanism (GAM) into the MDC) uncovers unseen spatial dependencies from traffic graph signals automatically 2. From the data stream, the multi-diffusion convolution (MDC) block harvests ST-features of the spatial domain. 3. The ST-TrafficNet, an LSTM based framework, harvests the features of the temporal domain. 4. The output from both ANN are summed up to achieve convolutional results. And 5. The ST-TrafficNet is evaluated on two benchmark datasets and compare it with various baseline methods for traffic forecasting.

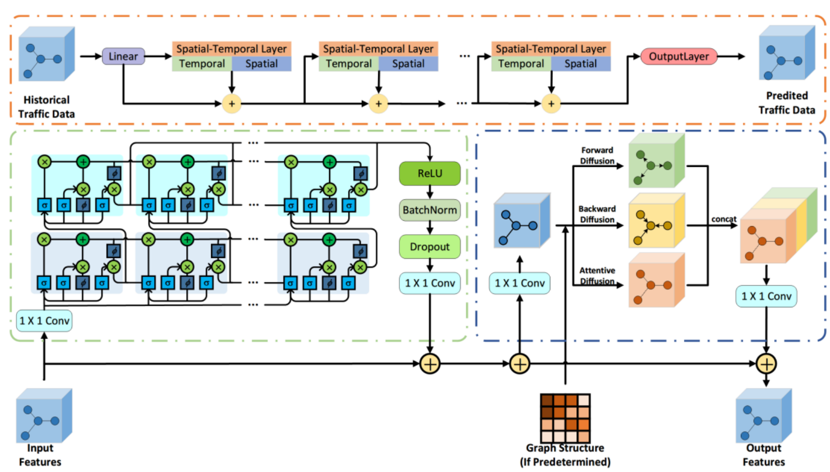


Fig 24. Lu's ST-TrafficNet architecture for traffic flow prediction [13]

Pan et al. [14] designed a deep learning framework for traffic flow prediction called "ST-Metanet" According to him, Traffic prediction is to enhance traffic safety and make the transportation system intelligent. However, it has to face to challenges: 1) complex spatio-temporal correlations of urban traffic and 2) diversity of such spatio-temporal correlations. To tackle these challenges, they proposed a deep-meta-learning based traffic model, entitled ST-MetaNet, to collectively predict urban traffic in all location at once. ST-MetaNet employs a seq2seq architecture, consisting of an encoder to learn historical traffic information and a decoder to make predictions step by step. More specifically, the encoder and decoder have the same network structure, which contains a recurrent neural network (RNN) to encode the urban traffic, a meta graph attention network (Meta-GAT) to capture diverse spatial correlations, and a meta recurrent neural network (Meta-RNN) to consider diverse temporal correlations. Extensive experiments were conducted based on two real-world datasets to illustrate the effectiveness of ST-MetaNet against several state-of-the-art methods.

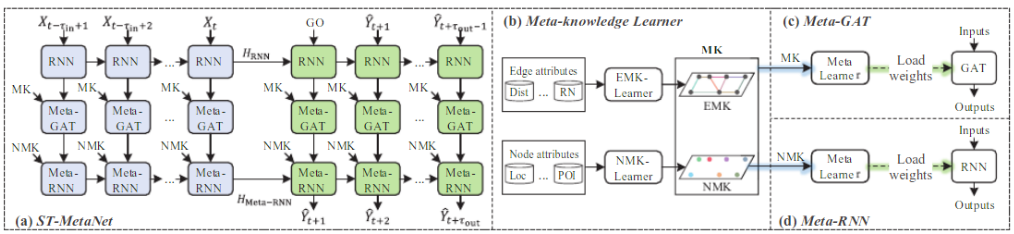


Fig 25. Pan's ST-MetaNet architecture for traffic flow & speed prediction [12]

De Medrano et al. [15] designed A Spatio-Temporal Spot-Forecasting Framework for Urban Traffic Prediction, named CRANN (Convo-Recurrent Attentional Neural Network). It is highly adaptable in several ST conditions, easy to understand and interpret, and better & more stable than state-of-the-art alternatives.

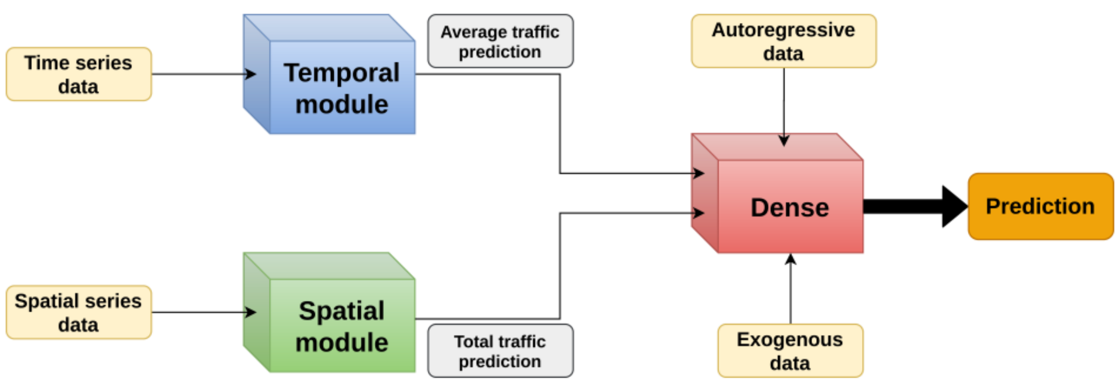


Fig 26. De Medrano's schematics for the CRANN architecture (a).

To cope the nonlinearity of the traffic flow data during the holidays, Luo et al, 2019 [16], designed a discrete Fourier transform (DFT) and support vector regression (SVR) based machine learning model to predict the road traffic flow during the holidays in Jiangsu Province, China, on Tomb-sweeping Day and National Day from 2011 to 2015. With proper training, the model outperformed other ML models – like ARIMA, SVR and EMD-SVR. The model is described in the paper itself [16].

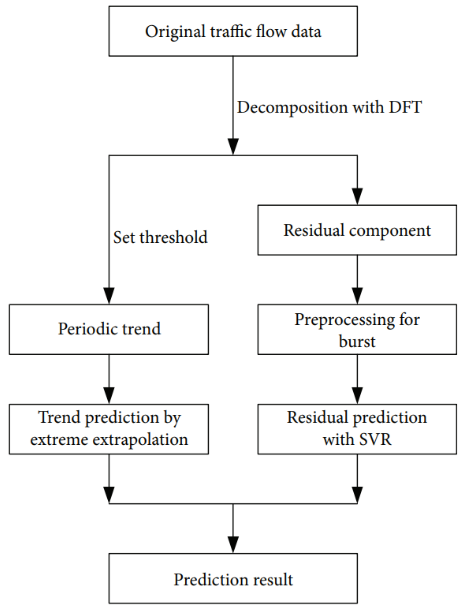


Fig 27. De Medrano's schematics for the CRANN architecture (b).

Shih et al. 2018 [18] designed an LSTM capable of processing multiple time series at the same time called "Temporal Pattern Attention LTSM" (TPA-LSTM). The architecture is designed to process complex and non-linear interdependencies between time steps of multivariate time series data. To obtain accurate prediction, an RNN with attention mechanism is designed and deployed to learn long-term dependency in time series data. The typical attention mechanism reviews the information at each previous time step and selects relevant information to help generate the outputs; however, it fails to capture temporal patterns across multiple time steps. The model uses a set of filters to extract time-invariant temporal patterns, like transforming time series data into its "frequency domain". The attention mechanism to select relevant time series and use its frequency domain information for multivariate forecasting. Surprisingly, regardless of the cases, the model achieved a comparable performance with other state-of-the-art models and architectures.

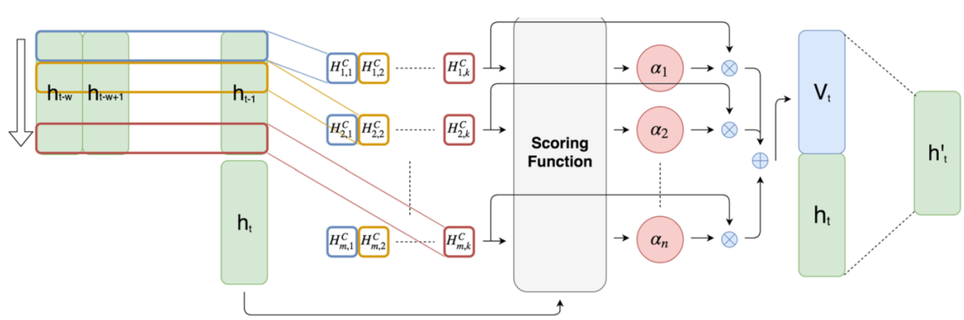


Fig. 2 Proposed attention mechanism. h_t represents the hidden state of the RNN at time step t. There are k 1-D CNN filters with length w, shown as different colors of rectangles. Then, each filter convolves over m features of hidden states and produces a matrix H^C with m rows and k columns. Next, the scoring function calculates a weight for each row of H^C by comparing with the current hidden state h_t. Last but not least, the weights are normalized and the rows of H^C is weighted summed by their corresponding weights to generate V_t. Finally, we concatenate V_t, h_t and perform matrix multiplication to generate h'_t, which is used to create the final forecast value (Color figure online)

Fig 28. Shih's Attentive RNN architecture for multivariate-time series prediction [18].

Atluri, Karpatne, and Kumar, 2017 [19], like Wang et al. in [8], restated and emphasized the concepts of spatiotemporal data mining.

2.2: Applications of Deep Learning Models in STDM

Restating what we have said in Chapter 1, the paper [8] and [19] cover several field of applications; such as climate science, neuroscience, social sciences, epidemiology, transportation, criminology, and Earth sciences. As our paper involve Geo-information system (GIS), we will only provide non-Neuroscience applications.

- **Meteorology:** Prediction of bad weather, as well as disasters, assist in reducing life and property loss,
  - For example, use of CNN is found in [1], [2], and [3]; as well as LSTM found in [6].
  - Zhang et al. 2018 [48], designed a Multilayer Perceptron Neural Network (MLPNN) for short term rainfall forecasting in China and achieved considerable performance – outperforming ARIMA and SVM.
- **Transportation:** Corresponding to fig 19, STDM has the most popular application in transportation.
  - With the rising availability of transportation data collected from various sensors, there is an urgent need to utilize deep learning methods to learn the complex and highly non-linear spatiotemporal correlations among the traffic data to facilitate various tasks.
  - Traffic flow, direction & speed prediction is the most popular application of this field, e.g. [7], [9], [12], [13], [14], [15], and [16].
- **On-Demand Service:** Taxi hailing apps are becoming popular. One of the notable works for taxi usage prediction is [10].
- **Human Mobility:** Mining the human mobility data is practically important for applications including traffic forecasting, urban planning, and human behavior analysis.
  - There are many recent works using deep learning methods for urban crowd flow prediction and crowd density estimation.
  - For example, Zhang et al, 2018 [49], designed an ST-ResNET based model to predict citywide crowd flow in Guiyang City, Beijing and New York City.
- **Location-Based Social Networks (LBSNs):** Some social network platforms, such as Foursquare and Flickr, use GPS features to locate the users and let the users broadcast their locations and other contents from their mobile device.
  - Currently, various deep learning models are designed to analyze the user generated ST data in LBSN.
  - For example, Zhao et al. 2018 [50], designed an ST-LSTM model to recommend POIs for Foursquare and Facebook users.
- **Criminology:** A criminal ST data consists of location coordinates, and timestamps.
  - DL models are set to predict crime incidence on a heatmap
  - For example, Duan et al. [51] proposed a Spatiotemporal Crime Network based on CNN to forecast the crime risk of each region in the urban area for the next day.

2.3: Summary

To sum up, most models referred, designed for traffic prediction work, are built with complex DL architectures, which improved accuracy and reliability over basic models. And we also categorized referred paper into several application categories, and discovered that the majority of them belong to traffic prediction category. DL-based architectures also contribute to data mining of different learning approaches, which are to be mentioned in section 3.1.

CHAPTER 3: METHODOLOGY

In this chapter, first we will list different approaches to STDM. Then, we will discuss the advantages and disadvantages of those approaches. After the discussion, different performance measures for STDM will be compared and we will try to address the effects of those measures. We will conclude this chapter with characteristics for different STDM.

3.1 Different approaches to spatial-temporal data mining

As the extension to Chapter 2, according to [8] and [19], ST-Dataset are mined using both traditional Machine learning methods – such as Decision tree, DBSCAN, SVM, ARIMA, etc. And deep learning methods – CNN, RNN, LSTM, Transformers, etc. There are some examples below.

- **Clustering:** Clustering refers to the grouping of instances in a data set that share similar feature values.
  - **In STDM, clustering can be performed on:**
    - Points, trajectories, time series, spatial maps, and ST rasters.
  - **There is a one challenge:** Clustering location based on their time series has to ensure that the discovered clusters are spatially contiguous. Ignoring this can lead to location data misinterpretation.
  - **Some notable algorithms:** ST-DBSCAN [20], clustering ST-points using DBSCAN algorithm [21], CLARANS [22], 'dynamic ST clusters' [23]. Etc.
- **Prediction:** Predictive learning learn a mapping from the input features to the output variables using a representative training set.
  - In spatio-temporal applications, both the input and output variables can belong to different types of ST data instances.
  - **Time series:** RNN and LSTM are widely used for time series data prediction. The weather variables such as wind speed are usually modeled as time series and then RNN/LSTM models are applied for future weather forecasting [24], [25], [26], [27], [28], [29].
  - **Spatial maps:** The spatial maps can be usually represented as image-like matrices, and thus are suitable to be handled by CNN for various predictive learning tasks [30], [31], [32], [33].
    - Zhang et al, 2016 [32], proposed a CNN based crowd flow forecasting model called UrbanFlow for real-time urban crow flow prediction.
    - To capture the temporal and spatial correlations of a sequence of spatial maps simultaneously, many works tried to combine CNN with RNN for the prediction. This method is utilized in [6], [7], [14], [15], and [18].
  - **Trajectories:** Currently, two types of deep learning models, RNN and CNN are used for trajectory prediction depending on the data representations of the trajectories.
    - Trajectories are a sequence type data of locations; which can be learned by an RNN or an LSTM
      - Can be also represented as a matrix, which can be learned by a CNN
      - Lv et al, 2018 [34], modelled trajectories as two-dimensional images, where each pixel represented trajectorial locations. Then an MLCNN were adopted to combine multi-scale trajectory patterns for destination prediction of taxi trajectories.
    - Some works also incorporate GPS, such as [35], who proposed a 4-layer LSTM model named DeepTransport, learning a set of GPS trajectories, to predict different transportation modes
  - **ST Raster:** ST raster can be represented as matrices whose two dimensions are location and time, or tensors whose three dimensions are cell region ID, cell region ID, and time.
    - Usually for ST Raster data prediction, 2D-CNN (matrices) and 3D-CNN (tensors) are applied, and sometimes they are also combined with RNN.
    - 3D-CNN models are purposed in [36], [37], [28] and [38].

        - Rasp & Lerch, 2018 [28], modelled the mobility events of passengers in a city in different time slots as a 3D tensor, and then used the 3D-CNN model to predict the supply and demand of the passengers for transportation.
        - The major difference between ST Raster and spatial map is that ST raster is the merged measurements in multiple time slots, while spatial map is the measurement in only one time slot.
    - **Using Temporal Information (Classic ML) [19]:** There are some estimation of time series' future trends
        - Exponential smoothing techniques [39], ARIMA models [40], and state-space models [41].
    - **Using Spatial Information (Classic ML) [19]:** There is a vast body of literature on spatial prediction methods that take into account the spatial auto-correlation structure in the data to ensure spatially coherent results.
        - Spatial auto-regressive (SAR) models [42], geographically weighted regression (GWR) models [43], and Kriging [44] Markov random field based approaches [45], [46], [47].
- **Representation Learning*:*** RL-models aims to learn the abstract and useful representations, formed by (non)linear transformations compositions of input data, of the input data to facilitate downstream data mining or machine learning tasks
    - Trajectories: Trajectories are ubiquitous in location-based social networks (LBSNs). CNN, GCN, GRU and RNN are used to learn such representations.
        - Ding et al, 2018 [52], proposed a geographical convolutional neural tensor network named GeoCNTN to learn the embeddings of the locations in LBSNs.
    - **Spatial maps:** There are several works that study how to learn representations of the spatial maps.
        - CNN, GCN, and Autoencoders are used in this learning method.
        - Costilla-Reyes et al, 2017 [53], proposed a CNN architecture for learning ST features from raw spatial maps of the sensor data.
- **Classification:** The classification task is mostly studied in analyzing fMRI data for disease identification.
    - In Neuroscience, deep learning models have become popular tools to analyze fMRI data for various classification tasks such as disease classification, brain function network classification and brain activation classification [54].
    - For instance [55] employed an LSTM to diagnose autism spectrum disorders (ASD) using fMRI time series data
    - [56], [57], [58], [59], [60], [61] modeled the fMRI data as spatial maps, and then used them as the input of the classification models.
    - Feature classification & recognition is also present in [3], [4], and [5], which used a CNN based architecture to classify features in satellite images.
- **Estimation and Inference:** Current works on ST data estimation and inference mainly focus on the data types of spatial map and trajectory.
    - Spatial map: While monitoring stations have been established to collect pollutant statistics, the number of stations is usually very limited due to the high cost.
        - One of the model utilizing this method is the ADAIN architecture, which [62] applied for modeling air quality inference of any location based on pollutants, and learning some complex feature interactions.
    - Trajectory: STDM of this purpose is used along with GPS
        - Which is notable in [10]
        - Zhang et al, 2018 [63], proposed a RNN based deep model named DEEPTRAVEL which can learn from the historical trajectories to estimate the travel time.
- **Anomaly Detection:** Anomaly detection or outlier detection aims to identify the rare items, events or observations that differ remarkably from the majority of the data.
    - ST-DBSCAN is used in ST point anomalies [64], [65] and [66].
    - Anomaly detection is also used in disaster analysis & prediction in [1], and [6]
    - For event,
        - DBN and LSTM is used on traffic accident tweets [67]

            - Zhu et al, 2018 [68], proposed to utilize Convolutional Neural Networks for automatically detecting the traffic incidents in urban transportation networks by using traffic flow data.
    - For spatial map,
        - ST-CNN is used in [69] for extreme weather events identification.
- **Other methods**
    - Change detection [19]
    - Pattern Mining [8]
    - Relation mining [8], [19]
    - POI recommender systems [8]
        - Wang et al, 2019 [70], proposed an attention based RNN for personalized route recommendation.
    - Etc.

3.2 Advantages and disadvantages of applying different approaches

Here, we will explain why deep learning models are preferred in recent works over traditional machine learning models, which can be seen more on Table 1:

	Deep Learning Models	Traditional ML models
ST Features	Automatic Learning	Hand-crafted
ST Data types	n-to-n	1-to-1
STDM Tasks	Prediction, classification, estimation, etc	Prediction, classification, estimation, clustering, frequent pattern mining, change detection, etc.
Temporal dependency	Long/Short term	Short term
Spatial dependency	Global/Local	Local
Interpretability	Low	High
Domain Knowledge	Little	Much

Table 1, comparison of DL vs ML methods in STDM. Source: [8].

According to table 1, we conclude that DL models are chosen over ML ones due to...

- **Automatic feature representation learning:** Unlike ML models; which can only handle hand-crafted ST-features, DL models can automatically learn hierarchical feature representations from the raw ST data
    - Multilayer convolution in CNN and recurrent structure in RNN can effectively learn highly complex ST-data
- **Powerful function approximation ability:** Each DLNN has multiple layers, which are nonlinear modules with convolution, pooling, dropouts, and activation functions.
    - With the composition of enough such transformations, very complex functions can be approximated to perform difficult STDM tasks with complex ST data.
- **Performing better with big data:** Traditional ML methods, e.g. SVM, Decision Tree, performs better on small datasets
    - But became ineffective when the dataset becomes larger
    - DL models, however, improve their performances when they learn a larger dataset.
        - Due to their powerful feature learning and function approximation abilities.

3.3 Different measures of performances

Usually, each DL framework has its performance measured in mean squared error (MSE), and root mean squared error (RMSE). For some models, however, use different performance metrics; such as that of Shih et al. 2018 [18], had some unfamiliar metrics, like RAE, RSE, CORR, etc. Thus his code needs an overhaul to make sure that input and output tensor dimensions are equal to each other – so that we can calculate MSE.

From each referred/surveyed paper, on Google™-Colab environment, we will run the Python codes written for each model and dataset. Then, we will run the models in the reference with their own datasets and compare them in the form of MSE & RMSE metrics – the less of them for each model, the better it performs in the term of learning and forecasting. Finally, we will conclude which model does the best learning/forecast. As this is a survey thesis, we will not design a custom model. The experiment is described in chapter 4.

3.4. Hypothesis:

3.1.1. STDM Deep learning models can do a variety of learning and predictions, but how well can they do?

MSE will tell us.

3.1.2. What architecture is the best model in STDM?

Is it true that ST-Metanet is the best model?

3.1.3. Do we need a custom architecture for such a task?

If we do, how can we design a DL-model?

3.1.4 What happens if we can modify the models' architectures to learn and process the same dataset?

3.5. Contributions

As you will see in chapter 4, we have studied several STDM papers and proved their model interesting and applicable. And we have run 3 models – Keras-LSTM, CRANN and ST-Metanet – all of them perfomed well for their architectures. More contributions can be read in Chapter 4.4.

CHAPTER 4: EXPERIMENTS

4.1. Experiment Settings

The list below are the potentially useful models in STDM, which are used in our comparative experiment.

Architecture	Dataset
LSTM-pytorch	Taxi-Uber Dataset (NYC, 2014-15)
LSTM-TF/Keras	Taxi-Uber Dataset (NYC, 2014-15)
CRANN	Madrid traffic & weather dataset (2018-19)
ST-Metanet (Flow)	Taxi-Flow (Beijing, 2010)
ST-Metanet (Speed)	METR-LA (Los Angeles, 2014)

Table 2. Models and datasets in the comparative experiment – brief table.

Continuing from section 3.3, the deep learning models and ANNs are written in Python with Pytorch, Mxnet & Tensorflow-Keras as our preferred modules. Be advised, not every model is runnable on the stock COLAB environment, and thus must be modified to meet their requirements. If a model consumes computing resources more than what COLAB can provide, it is advised to connect COLAB with a cloud VM instance, or a local Linux machine with RAM around 64 GB, for example, an EC2 Linux instance, plus a capable GPU.

Usually for each model, the dataset used to train it is the same as their respective original papers. The learning rate is 0.01. The optimiser is Adam's, and they are trained for 200 epochs. The main metrics, used to measure models' performance, are MSE and RMSE. Otherwise, the settings are specified to something else. Settings for every model are defined on table 3.

Architecture	Architecture Description	Dataset Description	Optimiser	Loss Function	learning rate	Epochs
Taxi-Simple-LSTM-pytorch	Simple-LSTM	Time series of Taxi-Uber DS. (2014-15)	Adam	MSE	0.01	200
Uber-Simple-LSTM-pytorch	Simple-LSTM	Time series of Taxi-Uber DS. (2014-15)	Adam	MSE	0.01	200
Taxi-Simple-LSTM-Keras	Simple-LSTM	Time series of Taxi-Uber DS. (2014-16)	Adam	MSE	0.01	200
Uber-Simple-LSTM-Keras	Simple-LSTM	Time series of Taxi-Uber DS. (2014-17)	Adam	MSE	0.01	200
CRANN-Temporal	Bahdanau Att.Mech	temporal time series of	Adam	MSE	0.01	200

	Autoencoder (LSTM based)	hourly/daily car traffic (in Madrid, 2018-19)				
CRANN-Spatial	CNN+ST-Att.Mech	incidence captured by 30 sensors + Timestamps (A 17000x30 matrix)	Adam	MSE	0.01	200
CRANN-Dense	Fully Connected Feedforward NN (FCFFNN)	dense 3D+ tensor of the both preceeding modules	Adam	MSE	0.01	200
ST-Metanet	Improved Seq2eq	Traffic flow & speed dataset	Adam	MSE	0.01	200
TPA-LTSM	Improved attentive lstm	muse dataset	Adam	tf.reduce.mean	1E-05	40
Lotto-Seq2seq	Bahdanau Attentive LSTM	UK lotto dataset (2016-2020)	Adam	sparse_categorical_crossentropy	0.0001	200

Table 3. Models and datasets in the comparative experiment – plus settings.

4.2. Preliminary Results

The yellow cells indicate the models requiring significant architectural changes to calculate the MSE due to unequal input-output tensor dimensions. The green characters indicate possible Metric values as if whose models were run (can be turned black later if a suitable environment is found). And the blue cells indicate models with memory problem – insufficient RAM – requiring a cloud environment to run. Despite stringent environmental restrictions preventing such models from running on COLAB, it is obvious that the model performed well, at least for its architecture. To see the detailed issues and possible solutions, see the section 4.3.

Architecture	**Key Metrics**			**Other Metrics**	
	MSE (either)	**RMSE(train)**	**RMSE(test)**	**Type**	**Value**
Taxi-Simple-LSTM-pytorch	75534780	8691.075	N/A		
Uber-Simple-LSTM-pytorch	0.03257497	0.1804854	N/A		
Taxi-Simple-LSTM-Keras	17995921.47	4242.16	10723.58		
Uber-Simple-LSTM-Keras	33913501.66	5823.53	10244.89		
CRANN-Temporal	6954.61768	-	83.39435	Rel.Err.%	9.2181
CRANN-Spatial	58546.6992	-	241.96425	Rel.Err.%	21.7391
CRANN-Dense	67154.6172	-	259.14208	Rel.Err.%	24.7697
seq2seq(flow)	1814.76	40.33592224	42.6	MAE	21.3
GAT-seq2seq(flow)	1267.36	33.13670731	35.6	MAE	18.3

ST-Metanet(flow)	1156	28.51646042	34	MAE	16.9
seq2seq(speed)	52.8529	6.668972492	7.27	MAE	3.55
GAT-seq2seq(speed)	44.3556	6.076465607	6.66	MAE	3.28
ST-Metanet(speed)	39.0625	5.799274921	6.25	MAE	3.05
TPA-LSTM	RAE=0.3118	RSE=0.4765	CORR=0.9850	Precision	0.58851
				Recall	0.68889
				F1	0.76333
Lotto-seq2seq	-	-	-	Sparse Top K	0.22571

Table 4. Experiment Results

To explain the results on Table 4, the simple LSTM models had a tremendous RMSE error in thousands due to simplicity of the model. Also due to overfitting, the UBER-LSTM Pytorch model is somewhat accurate. Nevertheless, we should never trust the number unless we see the graphs of figure 29 and 30. The TPA and the Lotto Seq2seq models are incomparable with the others due to metric difference.

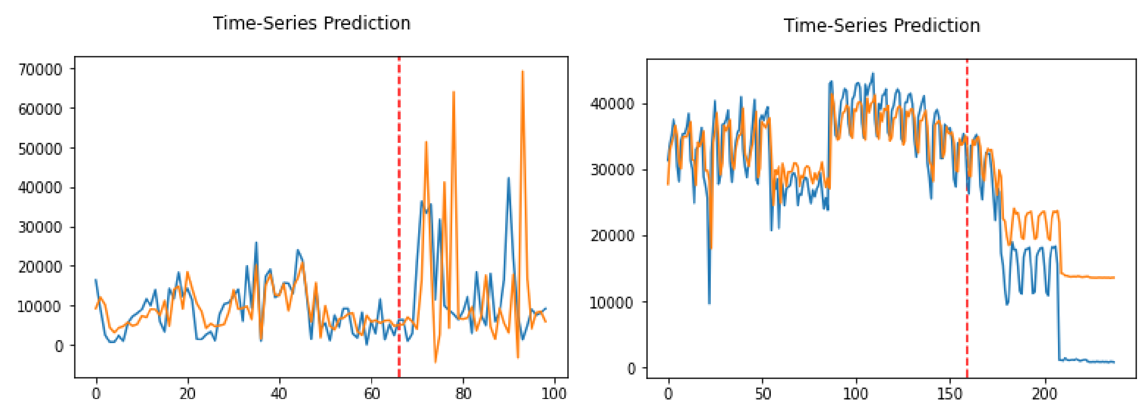


Figure 29: Pytorch-LSTM prediction results for (left) UBER and (right) TAXI ridership data

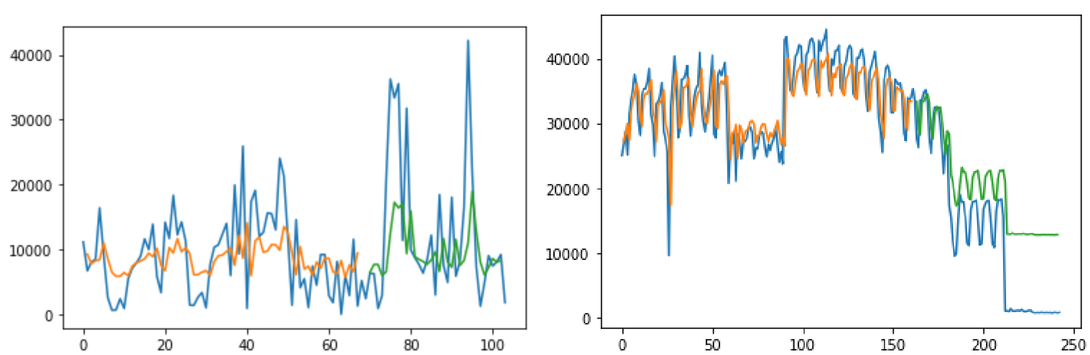


Figure 30: Keras-LSTM prediction results for (left) UBER and (right) TAXI ridership data

It is also apparent that complex state-of-the-art models (such as CRANN), when learning correct datasets, greatly outperformed simple models (like Keras-LSTM). As we can see on the table above, the ST-Metanet model, to date, is the best model for spatiotemporal traffic prediction due to complexity of the model containing MKL, GAT and RNN.

4.3 Issues:

The ST-Metanet model by Pan et al. [14] has a set of stringent environmental and modular requirements

- Which must be 100% met, otherwise the model will not run.
- Mxnet version worked: 1.4.0
    - But the notebook crashed due to insufficient RAM (required at least 32GB, or 64 GB at a safer level)
- Pymal has a 'six' dependency version conflict with pandas
- A Cloud VM seems to be the most promising solution
    - Otherwise connect a COLAB session to a Linux machine with 64GB RAM + a competent GPU.

The TRA-LSTM model by Shih et al. [18], despite suitable environment set on COLAB, takes forever to train.

- It is better to run it on a physical machine under Jupyter notebook, otherwise run it under COLAB PRO+ Environment.
    - Try connecting a COLAB session with a physical machine in the lab
    - Otherwise, run the model with Jupyter notebook in one of the lab machines

4.4 Current Assignments

- Study Attention Mechanisms
    - deeper
- Study APA & Chicago citation styles
    - deeper
- Find suitable environments for
    - ST-Metanet model of Mxnet-cu90 V.1.4.0
    - TPA-LSTM
- Run the Lotto-LSTM+Att model (of MSE)
- Rent a cloud VM of 64GB+1xGPU
- Run a spatial model for Taxiuber
    - Choose & implement a suitable model for Taxiuber spatial module (Likely a CNN based arch.)

- Reread papers to improve writing
    - deeper

4.5. Project Phases:

As this thesis is reaching its final phase, there is not much time for more experiments and must finish the paper by the end of December 2021.

- October& November
    - More Experiments
    - Environmental solution
    - Final Validation Runs
    - Statistical collection
    - Result & Analysis
    - Writing a paper.
- December
    - Finish the paper.
- January
    - Prepare for the defence oral exam

4.6. Accomplished Tasks (to be updated):

- Read at least 5 reference papers to study their models
- Run the following models
    - Lotto-LSTM+Att model (of the original metric)
    - Taxi-Uber LSTM
        - Pythorch
        - TF-Keras
    - ST-METANET (Insufficient RAM error)
    - CRANN
    - TPA-LSTM
- Run the taxi-uber LSTM time series forecasting model written in Keras
    - Model & coding are to be improved to extrapolate the future trends.
    - To implement a CNN/FCN-model for the spatial module is another task
- Write python notebooks for the ST-Metanet and the TPA-LSTM networks on colab
- CRANN – model deployment has been completed with satisfying results
- Add outline and technical basics to the paperwork
- Add explanation to the experiment & results section
- Add extra references found in [8] and [19]

CHAPTER 5: CONCLUSIONS

In this paper, we conduct a comprehensive overview of recent advances in machine and deep learning techniques for STDM. Also, we also performed a comparative experiment for each traffic predictive model. We first categorize the different data types and representations of ST data and briefly introduce the popular deep learning models used for STDM. Next, we raised some examples that use STDM techniques using deep neural networks. Then we overview recent works based on the STDM tasks including prediction, clustering, anomaly detection, estimation and inference, and others. Next we performed a comparative experiment of traffic prediction models, frameworks, and architectures to see which one produces least error in term of MSE and RMSE, concluding that each model did well for their respective architecture, and finding out that the ST-METANET [14], to date, performed the best learning and prediction. Finally, we list some open problems, discussing and point out the current issues, and the future research directions for this fast growing research filed.

REFERENCES

[1] Doshi, J., Basu, S. and Pang, G., 2018. From satellite imagery to disaster insights. *arXiv preprint arXiv:1812.07033*.

[2] Doshi, Jigar. 2018. Residual Inception Skip Network for Binary Segmentation. Pages 216–219 of:Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.

[3] Amit, S.N.K.B. and Aoki, Y., 2017, September. Disaster detection from aerial imagery with convolutional neural network. In 2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC) (pp. 239-245). IEEE.

[4] V. Iglovikov, S. Mushinskiy, and V. Osin, "Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition," vol. June, 2017.

[5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," ArXiv, vol. abs/2004.10934, 2020.

[6] Fang, Z., Wang, Y., Peng, L. and Hong, H., 2020. Predicting flood susceptibility using LSTM neural networks. *Journal of Hydrology*, [online] p.125734. Available at: <https://doi.org/10.1016/j.jhydrol.2020.125734> [Accessed 18 April 2021].

[7] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

[8] Wang, S., Cao, J., & Yu, P. (2020). Deep learning for spatio-temporal data mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

[9] Yu, B., Yin, H., & Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

[10] Correa, D., Xie, K., & Ozbay, K. (2017). Exploring the taxi and Uber demand in New York City: An empirical analysis and spatial modeling. In *96th Annual Meeting of the Transportation Research Board, Washington, DC*.

[11] Amato, F., Guignard, F., Robert, S., & Kanevski, M. (2020). A novel framework for spatio-temporal prediction of environmental data using deep learning. Scientific Reports, 2020(10), 22243. doi: 10.1038/s41598-020-79148-7

[12] Tang, Q., Yang, M., & Yang, Y. (2019). ST-LSTM: A Deep Learning Approach Combined Spatio-Temporal Features for Short-Term Forecast in Rail Transit. Journal of Advanced Transportation, 2019, Article ID 8392592. doi:https://doi.org/10.1155/2019/8392592

[13] Lu, H., Huang, D., Song, Y., Jiang, D., Zhou, T., & Qin, J. (2020). ST-TrafficNet: A Spatial-Temporal Deep Learning Network for Traffic Forecasting. Electronics, 2020(9), 1474.

[14] Pan, Z., Liang, Y., Wang, W., Yu, Y., Zheng, Y., & Zhang, J. (2019, July 25). Urban Traffic Prediction from Spatio-Temporal Data Using Deep Meta Learning. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '19: The 25th

ACM SIGKDD Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/3292500.3330884

[15] de Medrano, R., & Aznarte, J. L. (2020). A spatio-temporal attention-based spot-forecasting framework for urban traffic prediction. *Applied Soft Computing*, *96*, 106615.

[16] Luo, X., Li, D., & Zhang, S. (2019). Traffic flow prediction during the holidays based on DFT and SVR. *Journal of Sensors*, *2019*.

[17] Jiawei Han, Micheline Kamber, Jian Pei, 13 - Data Mining Trends and Research Frontiers, Morgan Kaufmann, 2012, Pages 585-631.

[18] Shih, S. Y., Sun, F. K., & Lee, H. Y. (2019). Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, *108*(8), 1421-1441.

[19] Atluri, G., Karpatne, A., & Kumar, V. (2018). Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys (CSUR)*, *51*(4), 1-41.

[20] Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering*, *60*(1), 208-221.

[21] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, No. 34, pp. 226-231).

[22] Ng, R. T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, *14*(5), 1003-1016.

[23] Chen, X., Faghmous, J. H., Khandelwal, A., & Kumar, V. (2015, June). Clustering dynamic spatio-temporal patterns in the presence of noise and missing data. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[24] Chen, M., Davis, J. M., Liu, C., Sun, Z., Zempila, M. M., & Gao, W. (2017, September). Using deep recurrent neural network for direct beam solar irradiance cloud screening. In *Remote Sensing and Modeling of Ecosystems for Sustainability XIV* (Vol. 10405, p. 1040503). International Society for Optics and Photonics.

[25] Cheng, L., Zang, H., Ding, T., Sun, R., Wang, M., Wei, Z., & Sun, G. (2018). Ensemble recurrent neural network based probabilistic wind speed forecasting approach. Energies, 11(8), 1958.

[26] Hossain, M., Rekabdar, B., Louis, S. J., & Dascalu, S. (2015, July). Forecasting the weather of Nevada: A deep learning approach. In 2015 international joint conference on neural networks (IJCNN) (pp. 1-6). IEEE.

[27] Liu, H., Mi, X., & Li, Y. (2018). Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. Energy Conversion and Management, 159, 54-64.

[28] Rasp, S., & Lerch, S. (2018). Neural networks for postprocessing ensemble weather forecasts. Monthly Weather Review, 146(11), 3885-3900.

[29] Zaytar, M. A., & El Amrani, C. (2016). Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. International Journal of Computer Applications, 143(11), 7-11.

[30] Ke, J., Yang, H., Zheng, H., Chen, X., Jia, Y., Gong, P., & Ye, J. (2018). Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services. IEEE Transactions on Intelligent Transportation Systems, 20(11), 4160-4173.

[31] Lee, D., Jung, S., Cheon, Y., Kim, D., & You, S. (2018). Forecasting taxi demands with fully convolutional networks and temporal guided embedding.

[32] Zhang, J., Zheng, Y., Qi, D., Li, R., & Yi, X. (2016, October). DNN-based prediction model for spatio-temporal data. In Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems (pp. 1-4).

[33] Zhu, Q., Chen, J., Zhu, L., Duan, X., & Liu, Y. (2018). Wind speed prediction with spatio–temporal correlation: A deep learning approach. Energies, 11(4), 705.

[34] Lv, J., Li, Q., Sun, Q., & Wang, X. (2018, January). T-CONV: A convolutional neural network for multi-scale taxi trajectory prediction. In 2018 IEEE international conference on big data and smart computing (bigcomp) (pp. 82-89). IEEE.

[35] Song, X., Kanasugi, H., & Shibasaki, R. (2016, July). Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (pp. 2618-2624).

[36] Chattopadhyay, A., Hassanzadeh, P., & Pasha, S. (2018). A test case for application of convolutional neural networks to spatio-temporal climate data: Re-identifying clustered weather patterns. arXiv preprint arXiv:1811.04817.

[37] Nguyen, N. T., Wang, Y., Li, H., Liu, X., & Han, Z. (2012, December). Extracting typical users' moving patterns using deep learning. In 2012 IEEE Global Communications Conference (GLOBECOM) (pp. 5410-5414). IEEE.

[38] Zhang, J., Zheng, Y., & Qi, D. (2017, February). Deep spatio-temporal residual networks for citywide crowd flows prediction. In Thirty-first AAAI conference on artificial intelligence.

[39] Gardner Jr, E. S. (2006). Exponential smoothing: The state of the art—Part II. International journal of forecasting, 22(4), 637-666.

[40] Box, G. E., & Jenkins, G. M. (1976). Time Series Analysis. Forecasting and Control (rev. ed.).

[41] Aoki, M. (2013). State space modeling of time series. Springer Science & Business Media.

[42] Kelejian, H. H., & Prucha, I. R. (1999). A generalized moments estimator for the autoregressive parameter in a spatial model. International economic review, 40(2), 509-533.

[43] Brunsdon, C., Fotheringham, S., & Charlton, M. (1998). Geographically weighted regression. Journal of the Royal Statistical Society: Series D (The Statistician), 47(3), 431-443.

[44] Oliver, M. A., & Webster, R. (1990). Kriging: a method of interpolation for geographical information systems. International Journal of Geographical Information System, 4(3), 313-332.

[45] Kasetkasem, T., & Varshney, P. K. (2002). An image change detection algorithm based on Markov random field models. IEEE Transactions on Geoscience and Remote Sensing, 40(8), 1815-1823.

[46] Schroder, M., Rehrauer, H., Seidel, K., & Datcu, M. (1998). Spatial information retrieval from remote-sensing images. II. Gibbs-Markov random fields. IEEE Transactions on geoscience and remote sensing, 36(5), 1446-1455.

[47] Zhao, Y., Zhang, L., Li, P., & Huang, B. (2007). Classification of high spatial resolution imagery using improved Gaussian Markov random-field-based texture features. IEEE Transactions on Geoscience and Remote Sensing, 45(5), 1458-1468.

[48] Zhang, P., Jia, Y., Gao, J., Song, W., & Leung, H. (2018). Short-term rainfall forecasting using multi-layer perceptron. IEEE Transactions on Big Data, 6(1), 93-106.

[49] Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., & Li, T. (2018). Predicting citywide crowd flows using deep spatio-temporal residual networks. Artificial Intelligence, 259, 147-166.

[50] Zhao, P., Zhu, H., Liu, Y., Li, Z., Xu, J., & Sheng, V. S. (2018). Where to go next: A spatio-temporal LSTM model for next POI recommendation. arXiv preprint arXiv:1806.06671.

[51] Duan, L., Hu, T., Cheng, E., Zhu, J., & Gao, C. (2017). Deep convolutional neural networks for spatiotemporal crime prediction. In Proceedings of the International Conference on Information and Knowledge Engineering (IKE) (pp. 61-67). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[52] Ding, D., Zhang, M., Pan, X., Wu, D., & Pu, P. (2018, April). Geographical feature extraction for entities in location-based social networks. In Proceedings of the 2018 World Wide Web Conference (pp. 833-842).

[53] Costilla-Reyes, O., Scully, P., & Ozanyan, K. B. (2017). Deep neural networks for learning spatio-temporal features from tomography sensors. IEEE Transactions on Industrial Electronics, 65(1), 645-653.

[54] Wen, D., Wei, Z., Zhou, Y., Li, G., Zhang, X., & Han, W. (2018). Deep learning methods to process fmri data and their application in the diagnosis of cognitive impairment: a brief overview and our opinion. Frontiers in neuroinformatics, 12, 23.

[55] Dvornek, N. C., Ventola, P., Pelphrey, K. A., & Duncan, J. S. (2017, September). Identifying autism from resting-state fMRI using long short-term memory networks. In International Workshop on Machine Learning in Medical Imaging (pp. 362-370). Springer, Cham.

[56] Guo, X., Dominick, K. C., Minai, A. A., Li, H., Erickson, C. A., & Lu, L. J. (2017). Diagnosing autism spectrum disorder from brain resting-state functional connectivity patterns using a deep neural network with a novel feature selection method. Frontiers in neuroscience, 11, 460.

[57] Heinsfeld, A. S., Franco, A. R., Craddock, R. C., Buchweitz, A., & Meneguzzi, F. (2018). Identification of autism spectrum disorder using deep learning and the ABIDE dataset. NeuroImage: Clinical, 17, 16-23.

[58] Horikawa, T., & Kamitani, Y. (2017). Generic decoding of seen and imagined objects using hierarchical visual features. Nature communications, 8(1), 1-15.

[59] Kim, J., Calhoun, V. D., Shim, E., & Lee, J. H. (2016). Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: Evidence from whole-brain resting-state functional connectivity patterns of schizophrenia. Neuroimage, 124, 127-146.

[60] Meszlényi, R. J., Buza, K., & Vidnyánszky, Z. (2017). Resting state fMRI functional connectivity-based classification using a convolutional neural network architecture. Frontiers in neuroinformatics, 11, 61.

[61] Shi, J., Zheng, X., Li, Y., Zhang, Q., & Ying, S. (2017). Multimodal neuroimaging feature learning with multimodal stacked deep polynomial networks for diagnosis of Alzheimer's disease. IEEE journal of biomedical and health informatics, 22(1), 173-183.

[62] Cheng, W., Shen, Y., Zhu, Y., & Huang, L. (2018, April). A neural attention model for urban air quality inference: Learning the weights of monitoring stations. In Thirty-second AAAI conference on artificial intelligence.

[63] Zhang, H., Wu, H., Sun, W., & Zheng, B. (2018). Deeptravel: a neural network based travel time estimation model with auxiliary supervision. arXiv preprint arXiv:1802.02147.

[64] Kut, A., & Birant, D. (2006). Spatio-temporal outlier detection in large databases. Journal of computing and information technology, 14(4), 291-297.

[65] Cheng, T., & Li, Z. (2004, June). A hybrid approach to detect spatial-temporal outliers. In Proceedings of the 12th International Conference on Geoinformatics Geospatial Information Research (pp. 173-178).

[66] Cheng, T., & Li, Z. (2006). A multiscale approach for spatio-temporal outlier detection. Transactions in GIS, 10(2), 253-263.

[67] Zhang, Z., He, Q., Gao, J., & Ni, M. (2018). A deep learning approach for detecting traffic accidents from social media data. Transportation research part C: emerging technologies, 86, 580-596.

[68] Zhu, L., Guo, F., Krishnan, R., & Polak, J. W. (2018, November). A deep learning approach for traffic incident detection in urban networks. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (pp. 1011-1016). IEEE.

[69] Racah, E., Beckham, C., Maharaj, T., Kahou, S. E., & Pal, C. (2016). ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. arXiv preprint arXiv:1612.02095.

[70] Wang, J., Wu, N., Zhao, W. X., Peng, F., & Lin, X. (2019, July). Empowering A* search algorithms with neural networks for personalized route recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 539-547).